CLAIMS

1. A method for determining whether to allow access to a protected resource from a server, comprising the steps of:

at a client, responsive to a request to retrieve the protected resource, generating a one-time only use piece of data which can be used to authenticate that the request is bound to a given identity contained in a cookie previously set by an authentication server;

forwarding the piece of data to the server in the request;

at the server, determining whether the piece of data is valid; and

if the piece of data is valid, executing an access control decision to determine whether to invoke the request.

2. The method as described in Claim 1 wherein the one-time only use piece of data is generated by applying a given function to a URL of the protected resource, a timestamp, a nonce generated by a server, the server's identity, and the client's identity.

3. The method as described in Claim 2 wherein the given function is a message authentication code (MAC)

calculated on the URL of the protected resource, the timestamp, the nonce, the server's identity, and the client's identity with a given key.

4. The method as described in Claim **3** wherein the given key is a symmetric key k_{sc} that binds the piece of data to the user identity contained in the identity cookie.

5. The method as described in Claim **4** wherein the symmetric key is generated by applying a one-way hash function to a shared client-server key k_c, the server identity, and a nonce.

6. The method as described in Claim **5** wherein the shared client-server key is generated by applying a one-way hash function to a user password.

7. The method as described in Claim **1** wherein the cookie includes a userid, the server identity, and a URL pointing to a location at the server that includes a script.

8. The method as described in Claim **1** wherein the cookie includes a userid, the server identity, and a URL

pointing to a location at the server that includes a script, and an access control token.

9. The method as described in Claim **8** wherein the script includes code for identifying whether a given piece of data is valid.

10. The method as described in Claim **9** wherein the script is accessed if the protected resource is located on a server other than the authentication server and the server and the authentication server are located within the same authentication domain.

11. A method of accessing a protected resource at a server, comprising the steps of:

at the server, receiving a request for a URL together with an identity cookie and a one-time only use authentication token associated with the request;

determining whether the authentication token is valid;

if the authentication token is not valid, returning to a requesting client an access denied message; and

if the authentication token is valid, executing an access decision function to determine whether to allow access to the protected resource.

12. The method as described in Claim **11** wherein the authentication token comprises a message authentication code (MAC) calculated on a URL of the protected resource, a nonce generated by the server, the server's identity, a user's identity, and a timestamp with a given key.

13. The method as described in Claim **12** wherein the given key is a symmetric key k_{sc} that binds the piece of data to the user identity as defined in the identity cookie.

14. The method as described in Claim **11** wherein the identity cookie includes a userid, an optional access control token, and a URL pointing to a location that includes a script.

15. The method as described in Claim **12** wherein the step of determining whether the authentication token is valid includes the steps of:

calculating a message authentication code;

evaluating whether the message authentication code is the same as the MAC in the authentication token.

16. The method as described in Claim **12** further including the step of saving the timestamp in a data structure to prevent replay of the authentication token.

17. The method as described in Claim **16** further including the step of saving a nonce generated by the server, the server's identity, and the user's identity to prevent replay of the authentication token by a client other than the user.

18. A computer program product in a computer-useable medium executable by a processor in a client computer, comprising:

code, responsive to a request to a server for retrieval of a protected resource, which generates a unforgeable piece of data which can be used at the server to authenticate that the request is bound to a given identity contained in a cookie previously set by an authentication server; and

code for inserting the piece of data into the request to the server.

19. The computer program product as described in Claim **18** further including a signed applet for installing the code in the client computer.

20. The computer program product as described in Claim **18** wherein the code which generates the unforgeable piece of data comprises:

code for calculating a message authentication code (MAC) on a URL of the protected resource, a nonce generated by a server, the server's identity, a user's identity and a timestamp with a given key.

21. The computer program product as described in Claim **20** wherein the code for calculating the message authentication code further includes code for generating the given key.

22. The computer program product as described in Claim **21** wherein the given key is a symmetric key that binds the piece of data to the user's identity contained in the cookie.

23. A computer program product for use in a computer-useable medium executable by a processor in a server, comprising:

code responsive to receipt of a request for a URL for a protected resource together with a one-time only use authentication token associated with the request for determining whether the authentication token is valid;

code for returning to a requesting client an access denied message if the authentication token is not valid; and

code for controlling execution of an access decision function if the authentication token is valid.

24. The computer program product as described in Claim **23** wherein the authentication token comprises a message authentication code (MAC) calculated on the URL of the protected resource, a nonce generated by a server, the server's identity, a user's identity, and a timestamp with a given key.

25. The computer program product as described in Claim **24** wherein the given key is a symmetric key k_{sc} that binds the piece of data to the user identity contained in an identity cookie set by an authentication server.

26. The computer program product as described in Claim **25** wherein the code for determining whether the authentication token is valid includes:

code for calculating a message authentication code; and

code for evaluating whether the message authentication code is the same as the MAC in the authentication token.

27. The computer program product as described in Claim **23** further including code for saving the timestamp and the authentication token in a data structure to prevent replay of the authentication token.

28. A method for issuing an access request from a client browser to a server hosting a protected resource, wherein an identity cookie has been set on the client browser by an authentication server, comprising:

using a symmetric key to derive a message authentication code (MAC) on a URL of the protected resource and a timestamp;

inserting the MAC together with the timestamp, the nonce set by the server, the server's identity, and a user's identity into a header of the request; and

forwarding the request to the server together with the identity cookie to enable the server to determine whether a requestor is authorized to access the protected resource.

29. The method as described in Claim **28** wherein a MAC is generated upon each request for the protected resource.

30. The method as described in Claim **28** wherein the symmetric key binds the MAC to a user identity contained in the identity cookie.

31. The method as described in Claim **30** wherein the symmetric key is generated by applying a one-way hash

function to a shared client-server key K_c, a nonce and the identity of the server that generated the nonce.

32. The method as described in Claim **31** wherein the shared client-server key is generated by applying a one-way hash function to a user password.

33. The method as described in Claim **28** wherein the identity cookie includes a userid, and a URL pointing to a location that includes a script.

34. The method as described in Claim **28** wherein the identity cookie includes a userid, a URL pointing to a location that includes a script, and an access control token.

35. The method as described in Claim **34** wherein the script includes code for identifying whether a MAC is valid.